

**IN THE SPECIFICATION**

Please replace the paragraph beginning at page 98, line 25, with the following amended version:

--PMU 5001 has hardware-controlled local packet memory (LPM) 5006. LPM 5006 is the memory where packets reside during SPU processing. A register transfer unit (RTU) 5007 is provided within PMU 5001 and is adapted by hardware to select and load, and to transfer contexts between PMU 5001 and SPU 5003 ownership. RTU 5007 has logic (L) ~~5008 5009~~ implemented thereon and adapted to enable context selection and activation according to an object of the present invention. Bi-directional arrows illustrated within PMU 5001 illustrate bi-directional communication between illustrated components.--

Please replace the paragraph beginning at page 100, line 10, with the following amended version:

--In practice of the invention in a preferred embodiment, logic L ~~(5008 5009)~~ within RTU 5007 has a determination trigger that executes a selection and release sequence when it is detected that all contexts are PMU-owned. In one embodiment, RTU ~~5008 5007~~ consults a table of context states at specific points in time which may be every cycle, every second cycle, or once every few cycles. When the table indicates that all of contexts 5005 are PMU-owned, the rest of the logic function is triggered to execute.--

Please replace the paragraph beginning at page 100, line 17, with the following amended version:

--Upon detecting that all contexts are PMU-owned, L ~~5008 5009~~ must then determine if there are any idle contexts ~~5008 5005~~ that are not being pre-loaded with packet data for normal processing. If there is one context sitting idle, then L ~~5008 5009~~ selects that context for release. If there are more than one context sitting idle, then L ~~5008 5009~~ either randomly selects one, or selects one based on a priority scheme. A selected context will have a higher priority than a non-selected context. This priority selection logic may be integrated with logic normally used in selecting a best available context for processing data packet information. For example, if more than one idle context exists in the event that all

are PMU-owned, then a priority would be to select one wherein access to all required functional units is likely to be unfettered. In this way, a potentially high priority interrupt can be serviced using fewer cycles. This part of the selection logic may be programmed to look for a context that would not have to compete with several other contexts for access to functional units. It is noted herein that such a priority determination would have to be predictive because at the time of selection, no other contexts are presumed to be SPU-owned.--

Please replace the paragraph beginning at page 101, line 13, with the following amended version:

--It will be apparent to one with skill in the art that in the event that all contexts 5005 are PMU-owned, it may be that there are none idle. That is, it may be true that all have been selected and are being loaded. In this case, logic ~~5008 5009~~ may abort the selection process and look for a next opportunity when all contexts are PMU-owned and an idle context exists. In an alternative embodiment, it is simply not allowed that all contexts 5005 may, at a same instant, be PMU-owned. In this case, a context is isolated and dedicated to listen for pending interrupts. Also in this case, the dedicated context will not be used for normal processing, and is a separate and dedicated resource. In this case, logic would have to be implemented within core 5004 and within RTU 5007. The cost of implementation plus the performance penalty taken by having fewer contexts to perform packet processing may therefore outweigh any real benefit of dedication. This assumes also that all contexts 5005 are programmable for dedication as a SPU-owned context for running interrupt service routines.--

Please replace the paragraph beginning at page 101, line 13, with the following amended version:

--FIG. 51 is a process flow chart illustrating logic steps for selecting and releasing a context according to an embodiment of the present invention. At step 5100, the logic ~~5008 5009~~ (FIG. 50) determines if all contexts are PMU-owned. This step may be performed aggressively by periodically consulting context state information. A period may be every one or a few cycles. Step 5100 may also be performed passively through notification. For example, a context-state machine (software table) may automatically trigger the logic in the event that all contexts become PMU-owned.--